3. Error Correcting Codes

References V. Bhargava , “ Forward Error Correction Schemes for Digital Communications”, IEEE Communications Magazine, Vol 21 No1 pp11 – 19, January 1983
Mischa Schwartz, “Information Transmission Modulation and Noise”, 4th ed, Chapter 7, Mc Graw Hill 1990
Shu Lin, Daniel J Costello, “Error Control Coding: Fundamentals and Applications”, Prentice Hall, 1983
Irving S Reed, Xuemin Chen, “Error-Control Coding for Data Networks”, Kluwer Academic, 1999
Robert Morelos-Zaragoza, “The Art of Error Correcting Coding”, 2nd ed., Wiley 2006

Introduction

In statistical communication theory, using binary PSK transmission, over a binary symmetric channel, BSC, the probability of a bit error event over an additive white Gaussian noise channel, (AWGN)is given by

$$P_e = \frac{1}{2} erfc\sqrt{\frac{E_b}{N_0}} \qquad (3.1)$$

where E_b is the symbol energy and n_0 is the spectral noise energy, and the erfc(x) is the complementary error function
erfc(x) = 1 – erf(x) where

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt \qquad erfc(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}dt \qquad (3.2)$$

This is also given in terms of

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \qquad \text{where} \qquad Q(x) = \frac{1}{2}erfc(\frac{x}{\sqrt{2}}) \qquad (3.3)$$

The classical Digital Communication System is shown in Fig 3.1. In this section we are looking at the channel encoding and decoding in the presence of noise that can be additive or burst noise. The channel can be any type from cable to satellite, as well as electromagnetic storage involving CD’s and other data storage media.

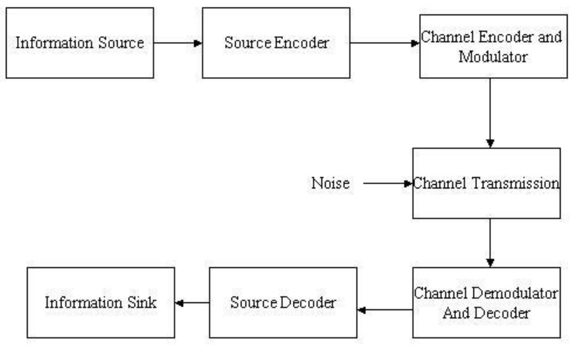


Figure 3.1

Basic jargon associated with error correcting codes

(i) a *block code* can be of various types. But all types have a block of *k data bits* associated to a *codeword of n bits*. This is called an (n, k) code. The *code rate*, which indicates the efficiency of the code, is k/n and is always a fraction.

(ii) A binary code's capability is defined in terms of the XOR outcome between any two of its codewords. An (n,k) code has 2^k *codewords* spread over a vector space of 2^n *n-tuples*. The outcome is called the distance between two codewords. The minimum distance of a code is the *minimum distance* between any two codewords, denoted by t.

(iii) The error correcting capability in bits, given by d, is obtained as

$$d \leq 2t + 1$$

So a code with minimum distance 3, can correct 1 bit in error.

(iv) There are also codes that have memory over the blocks, called *convolutional codes*. These type of codes are defined on 3 parameters, (n, k, r) where r denotes the number of codewords being generated that are influenced by any block of data being used.

Clearly there must be a mechanism of how to obtain codes, based on some technique that gives the best, or good, minimum distance. Further for large k and n there must be techniques of generating automatically the codewords, and not have a memory through which to compare and associate the current data to its appropriate codeword. There must also be techniques of automatically checking the received n-tuple to be able to decide on whether it is correct, and if it is not correct, to automatically correct the error. Linear Algebra theory based on *groups*, *fields*, *Galois Fields* of the type GF(2) and GF(2^m), is used extensively as a basis for generating efficiently good error correcting codes.

The above is based also on the premise that the received n-tuple does not have error bits that exceed the capabilities of the code, as this may result in *undetectable errors*. These are to be avoided as much as possible, (in fact in some applications at all costs), and therefore application type, signal power, and expected channel noise, have to be analysed to decide on the necessary error code capability, before deciding on which code to use. Further there are applications where two types of codes are used to increase the robustness of the coding.

Block Codes

A Linear Block Codes

In a linear block code the sum of two codewords, based on XOR, results in another codeword. This means that the codewords form a closed subspace over the field of GF(2). Initially we will consider the necessary algebra to understand the GF(2).

(a) Definition of a **group**:
A set of elements G with an operation * is called a group if:

(i) the binary operation U* is associative;
(ii) G contains an element e, the identity element, such that
A*e = e*a = a
(iii) for any element a, there exists an inverse element a' such that
a*a' = a'*a = e

A group is commutative if for any two elements a, b
A*b = b*a
The binary set 0,1 is a group under the operation of XOR. The integers 0 to (N-1) are a group under modulo(N) addition.
The integers {1,2,…,p} are also a group under modulo (N) multiplication IF p is prime.

	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Table of integer modulo-5 under multiplication

(b) Definition of a **field**
In this case there are two operators and in general one can talk about addition, subtraction, multiplication and division, where operation on two elements of a set {F} still results in another element of set {F}

(i) F is commutative under addition, (+), and the identity element is 0.
(ii) F is commutative under multiplication (.). The identity elemnt is the unit element , 1
(iii) Multiplication is distributive over addition
a.(b+c) = a.b + a.c

A field with a finite number of elements is called a *finite field.* Finite fields are also called Galois Fields. Galois Fields with a number of elements that is prime, in particular GF(2) play an important role in coding theory.

Modulo-2 Addition		
+	0	1
0	0	1
1	1	0

Modulo-2 Multiplication		
.	0	1
0	0	0
1	0	1

GF(2)

Sets of integers {0,1,2,…,p-1} of p elements, where p is a prime constitute a **Galois Field** GF(p).

Note that the process of addition involves adding the inverse element under addition, while division involves multiplying the inverse element under multiplication.
An extension of GF(p, p a prime, to a GF(p^m) is also possible. We will be making extensive use of Galois fields of the type GF(2^m).

A linear block code is an (n,k) code where every codeword is the modulo-2 sum of any other two codewords.

A linear block code is characterized by a *generator matrix*. The generator matrix is made up of a parity check and an Identity matrix.
Ex. For a (7,4) code, the generator matrix consists of

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \vdots & 1 & 0 & 0 & 0 \\ p_{10} & p_{11} & p_{12} & \vdots & 0 & 1 & 0 & 0 \\ p_{20} & p_{21} & p_{22} & \vdots & 0 & 0 & 1 & 0 \\ p_{30} & p_{31} & p_{32} & \vdots & 0 & 0 & 0 & 1 \end{bmatrix}$$

a parity check [4 x 3] matrix followed by an Identity [4 x 4] matrix to make up a [4 x 7] matrix. In this format the linear block code is also called a systematic code since the structure of the code separates out the derived parity checks and the original data block into separate subblocks.

Example: Linear Block Code (7,4) has a generator matrix

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 111 & 1000 \\ 101 & 0100 \\ 011 & 0010 \\ 111 & 0001 \end{bmatrix}$$

and results in the $2^4 = 16$ codewords listed below

Data	Codeword
0000	000 0000
0001	111 0001
0010	011 0010
0011	100 0011

0100	101 0100
0101	010 0101
0110	110 0110
0111	001 0111
1000	110 1000
1001	001 1001
1010	101 1010
1011	010 1011
1100	011 1100
1101	100 1101
1110	000 1110
1111	111 1111

The *Hamming Weight* of every codeword is at least three. The minimum distance of the code is the minimum difference between any two codewords, but since any codeword is the sum of any other two codewords, the minimum *Hamming distance* is 3.

The *parity-check matrix* of an (n,k) linear block code is derived from the [k x n] generator matrix, as an [(n-k) x n] matrix given by

$H = I^{n-k}.P^T$ where I is the identity matrix and P the parity-check part of the generator matrix.

For the (7,4) code above, the parity-check matrix H is a [3 x 7] matrix, given by

$$H = \begin{bmatrix} 1\,0\,0 & 1\,1\,0\,1 \\ 0\,1\,0 & 1\,0\,1\,1 \\ 0\,0\,1 & 0\,1\,1\,1 \end{bmatrix}.$$

The importance of the parity-check matrix is due to the fact that

$G.\ H^T = 0$; also implies that $v.\ H^T = 0$; where v is any valid codeword

Syndrome and Error Detection

In general, when a received word is passed through the H^T matrix it gives rise to an [n-k] bit pattern called the syndrome. A non-zero syndrome indicates an error has been detected.
One can write the received vector as **r = e + v** where **r** is the received n-tuple that can be considered to be made up of a codeword **v** and an error vector **e**.

However there is still the possibility of an undetectable error if the error vector **e** alters one codeword to another. There are $(2^k - 1)$ received n-tuples that can be undetectable errors. The rest, i.e. $2^n - 2^k$ are detectable.
The probability of an undetectable error, denoted by $P_u(E)$ is given by

$$P_u(E) = \sum_{i=1}^{n} A_i\, p^i (1-p)^{n-i} \qquad (3.4)$$

where A is the Hamming weight of each codeword and

$p \equiv p_e$ is the probability of bit error, (transition error probability) on a BSC.

And $(1-p) \equiv p_c$ is the probability of a correctly received bit

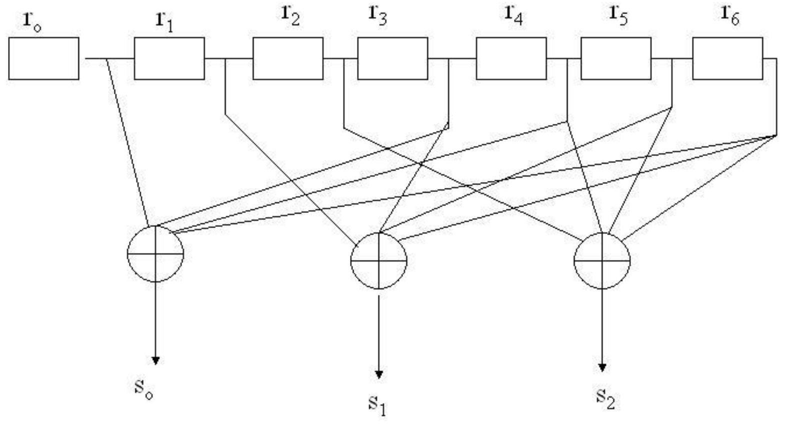
For the (7,4) code above the weight distribution is $A_0 = 1$, $A_3 = A_4 = 7$, $A_1 = A_2 = A_5 = A_6 = 0$. Therefore

$$P_u(E) = 7p^3(1-p)^4 + 7p^4(1-p)^3 + p^7$$

and for a $p = 10^{-2}$, the $P_u(E) = 7 \times 10^{-6}$, considerably lower than p.

Syndrome circuit

From H^T of the (7,4) code above the syndrome bits are
$s_0 = r_0 + r_3 + r_4 + r_6$; $s_1 = r_1 + r_3 + r_5 + r_6$; $s_2 = r_2 + r_4 + r_5 + r_6$; and they can be obtained from an XOR logic circuit given below.



The next important issue is whether given a syndrome, and the consequent indication of an error, there is the possibility of automatically correcting the error. For a (7,4) code there are $2^7 - 2^4 = 128$ detectable error patterns, that are detected by $(2^3-1) = 7$ non-zero syndrome patterns. Clearly there is a mapping of 16 to 1.
Using maximum likelihood, the error vector with the minimum Hamming weight from the set of 16, should be chosen. Denoting this vector by **e,** the resultant codeword would be

$$\mathbf{v} = \mathbf{r} + \mathbf{e}$$

The 16 patterns that are a set for each syndrome pattern, together with the sixteen codewords when the syndrome is zero, form the $2^7 = 128$ possible 7-bit tuples for the (7,4) code. For a linear block code these give rise to a standard array. The pattern with the minimum Hamming weight in each of the eight cosets, is used as a coset leader.

0000000	1110001	0110010	10000011	1010100	0100101	1100110	0010111	1101000	0011001	1011010	0101011	0111100	1001101	0001110	1111111
1000000	0110001	1110010	00000011	0010101	1100101	0100110	1010111	0101000	1011001	0011010	1101011	1111100	0001101	1001110	0111111
0100000	1010001	0010010	1100011	1110100	0000101	1000100	0110111	1001000	0111001	1111010	0001011	0011100	1101101	0101110	1011111
0010000	1100001	0100010	1010011	1000100	0110101	1110110	0000111	1111000	0001001	1001010	0111011	0101100	1011101	0011110	1101111
0001000	1111001	0111010	1001011	1011100	0101101	1101110	0011111	1100000	0010001	1010010	0100011	0110100	1000101	0000110	1110111
0000100	1110101	0110110	1000111	1010000	0100001	1100010	0010011	1101100	0011101	1011110	0101111	0111000	1001001	0001010	1111011
0000010	1110011	0110000	1000001	1010110	0100111	1100100	0010101	1101010	0011011	1011000	0101001	0111110	1001111	0001100	1111101
0000001	1110000	0110011	1000010	1010101	0100100	1100111	0010110	1101001	0011000	1011011	0101010	0111101	1001100	0001111	1111110

Standard Array for the (7,4)code

The coset leaders are in the first column of each coset and are the 7-bit pattern with minimum weight in that coset.

Finally, the relationship of the syndrome to the coset leaders is given by

Syndrome	Coset Leader
100	10000000
010	0100000
001	0010000
110	0001000
101	00000100
011	00000010
111	00000001

Note that the order of the syndrome patterns correspond to the row order of the H^T . For a k greater than 8, the size of the table becomes considerable, and the search for the corresponding error vector to add to the received 7-bit pattern to obtain the codeword becomes long.

There is a relationship between a generator matrix and its parity check matrix in terms of dual codes. If H is used for code C, then the same H is the G for the dual code $C_{d.}$ This can be used to calculate the $P_u(E)$ when it is not easy to get the codewords weight distribution A_i of the code because k is big. Defining the two weight distributions in terms of polynomials as

C $\quad A(z) = A_0 + A_1z + \ldots + A_nz^n$ and for its dual (3.5)

$C_d \quad B(z) = B_0 + B_1z + \ldots + B_nz^n$ where z is given by (3.6)

z = p/(1-p) and A(z) can be obtained in terms of B(z) from

$$A(z) = 2^{-(n-k)}(1+z)^n B(\frac{1-z}{1+z}) \qquad (3.7)$$

and using (3.4) given by $P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i}$, by rearranging to obtain

$$P_u(E) = (1-p)^n \sum_{i=1}^{n} A_i (\frac{p}{1-p})^i \qquad (3.8)$$

and since $A_0 = 1$, then using the two equations (3.7) and (3.8), an alternative to (3.4) can be obtained as

$$P_u(E) = 2^{-(n-k)} B(1-2p) - (1-p)^n \qquad (3.9)$$

where $B(1-2p)=\sum_{i=0}^{n} B_i(1-2p)^i$ (3.10)

Either 3.4 or (3.9) can be used to obtain $P_u(E)$ depending on which of k or (n-k) is suitable.

Example: For the (7.4) code having G and H above, the C_d is a (7,3) code whose G is given by $G=\begin{bmatrix}1\,0\,0 & 1\,1\,0\,1\\ 0\,1\,0 & 1\,0\,1\,1\\ 0\,0\,1 & 0\,1\,1\,1\end{bmatrix}$ and the eight codewords are given by

000 0000	001 0111	010 1011	011 1100	100 1101
101 1010	110 0110	111 0001		

The weight distribution gives $B(z) = 1 + 7z^4$

Hence from the dual code

$P_u(E)=2^{-3}[1+7(1-2p)^4]-(1-p)^7$ which is equal to the previous answer. (*Check it out by working out the polynomial in p for both cases).

In general, for large (n.k) it is not easy to work out $P_u(E)$. It can be shown that an upper bound exists given by

$P_u(E)=2^{-(n-k)}[1-(1-p)^n]$ and since $[1-(1\text{-}p)^n]$ is ≤ 1 then $P_u(E) \le 2^{-(n-k)}$

Hamming Codes

These are linear codes based on the following properties. For any $m \ge 3$, a Hamming code can be built with

$n = 2^m - 1$; $k = 2^m - m - 1$; $n\text{-}k = m$.

Note that the (7,4) code above is a Hamming code. One particular property, because of the nature of the number of syndrome patterns available, is that a Hamming code is a perfect code. The class of t-error correcting codes that has in its standard array all patterns of t or less weight as the coset leaders, is called a perfect code.

Perfect codes are rare. Besides the Hamming Codes, the other nontrivial perfect code is the (23,12) Golay code.

The Hamming code therefore cannot detect two or more errors. However this can be done using a shortened Hamming code obtained by removing from the H matrix all the columns whose weight is even. If l columns are removed, the code now becomes

Code length	$n = 2^m - l - 1 = n - l$
Information bits	$k = 2^m - m - l - 1$
Parity check bits	$n\text{-}k = m$
Minimum distance	$d \ge 3$

Using this shortened cyclic code decoding is done as follows:

1. If the syndrome is zero a correct codeword is received
2. If the syndrome is odd, then a single error occurred which can be corrected
3. If the syndrome is even there is an uncorrectable error pattern